Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application:

1. (Currently Amended) A method of storing data pages at a proxy, the method comprising:

a) receiving a data page, wherein said data page includes~~;~~

~~b) receiving~~ page dependency data that contains one or more dependencies such that each dependency indicates an underlying data source which said data page is dependent on;

~~c)~~ b) storing said data page;

~~d)~~ c) storing said page dependency data;

~~e)~~ d) receiving an event;

~~f)~~ e) determining if said event changes an underlying data source corresponding to one of said page dependency data ~~associated with said data page~~; and

~~g)~~ f) updating the cache by refreshing or deleting said data page if said event changes ~~an underlying data source corresponding to~~ one of the page dependency data ~~associated with said data page~~.

2. (Original) A method as defined in claim 1, where said page dependency data are written in HTML or XML.

3. (Original) A method as defined in claim 1, where said page dependency data are generated by a Request-Based dependency generator.

4. (Original) A method as defined in claim 3, where said Request-Based dependency generator uses a URL request of said data page.

5. (Original) A method as defined in claim 3, where said Request-Based dependency generator uses a configuration file to generate said page dependency data.

6. (Original) A method as defined in claim 1, where said page dependency data are generated by a script-based dependency generator.

7. (Original) A method as defined in claim 6, where said page dependency data are encoded in said data page by a script-based dependency generator.

8. (Original) A method as defined in claim 1, where said page dependency data are manually encoded into a data file.

9 (Original) A method as defined in claim 1, where said data page and said page dependency data are stored in one or more files.

10. (Original) A method as defined in claim 1, where said storing of said data page at the proxy is in response to data in a configuration file.

11. (Canceled)

12. (Previously Presented) A method as defined in claim 1, where send event is received incorporated in an event message.

13. (Previously Presented) A method as defined in claim 1, where said event is written in HTML or XML.

14. (Previously Presented) A method as defined in claim 1, where said event came from a Request-Based event generator.

15. (Original) A method as defined in claim 14, where said Request-Based event generator uses a configuration file.

16. (Original) A method as defined in claim 14, where said Request-Based event generator uses a URL request of said data page.

17. (Original) A method as defined in claim 16, where said URL request is parsed to obtain parameters.

18. (Original) A method as defined in claim 16, where said tJRL request includes request header information.

19. (Original) A method as defined in claim 18, where said tJRL request is parsed to obtain parameters.

20. (Previously Presented) A method as defined in claim 1, where said event came from a script-based event generator.

21. (Previously Presented) A method as defined in claim 1, where said event came from a trigger-based event generator.

22. (Previously Presented) A method as defined in claim 1, where said event came from polling event generator.

23. (Previously Presented) A method as defined in claim 1, where said event came from a custom event generator.

24. (Previously Presented) A method as defined in claim 1, where said determination if said event changes one of the page dependency data associated with said data page is done by a change event evaluator.

25. (Previously Presented) A method as defined in claim 1, where said determination by the change event evaluator is done by matching said page dependency data with said event.

26. (Previously Presented) A method as defined in claim 1, where said event was generated by Request-Based event generator and sent to a change event evaluator.

27. (Previously Presented) A method as defined in claim 1, where said updating the cache involves keeping the index of tJRL addresses and page dependency data up-to-date.

28. (Currently Amended) A computer software product for use in a computer system that executes program steps recorded in a computer-readable media to perform a method for enabling storing of data pages at a proxy comprising:

a) a recordable media; and

b) a program of computer-readable instructions executable by the computer to perform method steps comprising:

i) receiving a data page, wherein said data page includes;

~~ii)~~ ~~receiving~~ page dependency data that contains one or more dependencies such that each dependency indicates an underlying data source which the said data page is dependent on;

~~iii)~~ ii) storing said data page;

~~iv)~~ iii) storing said page dependency data;

~~v)~~ iv) receiving an event;

~~vi)~~ v) determining if said event changes said one of the page dependency data associated with said data page; and

~~vii)~~ vi) updating the cache by refreshing or deleting said data page if said event changes said one of the page dependency data associated with said data page.

29. (Currently Amended) A proxy server system that provides stored data files without requesting data files from the origin web server, comprising:

a) a central processing unit that can establish communication with a user computer;

b) a storage device;

c} a processor connected to the storage device wherein the storage device stores:

d) the processor is operative with said program component to:

i) receive a data page, wherein said data page includes;

~~ii) receive~~ page dependency data that contains one or more dependencies such that each dependency indicates an underlying data source which the said data page is dependent on;

~~iii)~~ ii) store said data page;

~~iv)~~ iii) store said page dependency data;

~~v)~~ iv) receive an event;

~~vi)~~ v) determine if said event changes said one of the page dependency data associated with said data page; and

~~vii)~~ vi) update the cache by refreshing or deleting said data page if said event changes said one of the page dependency data associated with said data page.

30. (Currently Amended) A method for updating a data page in a cache, comprising:

receiving an event indicating a change in a data source;

examining dependency data located inside the data page to determine if the data page

depends on said data source; and

updating the data page if the data page depends on said data source.

31. (Previously Presented) The method of claim 30, wherein said updating includes deleting the data page from the cache if said event indicates that said data source has been deleted.

32. (Previously Presented) The method of claim 30, wherein said updating includes replacing the data page with an updated version of the data page if the event indicates that said data source has been updated.

33. (Currently Amended) An apparatus for updating a data page in a cache, comprising:

means for receiving an event indicating a change in a data source;

means for examining dependency data located inside ~~for~~ the data page to determine if the data page depends on said data source; and

means for updating the data page if the data page depends on said data source.

34. (Previously Presented) The apparatus of claim 30, wherein said means for updating includes means for deleting the data page from the cache if said event indicates that said data source has been deleted.

35. (Previously Presented) The apparatus of claim 30, wherein said means for updating includes means for replacing the data page with an updated version of the data page if the event indicates that said data source has been updated.

36. (Currently Amended) A program storage device readable by a machine, tangibly embodying a program of instructions executable by the machine to perform a method for updating a data page in a cache, the method comprising:

receiving an event indicating a change in a data source;

examining dependency data located inside ~~for~~ the data page to determine if the data page depends on said data source; and

updating the data page if the data page depends on said data source.